Stabilizing cluster structures in mobile networks for OLSR and WCPD as Basis for Service Discovery

Tom Leclerc, Adrian Andronache, Laurent Ciarletta, Steffen Rothkugel

Stabilizing cluster structures in mobile networks for OLSR and WCPD as Basis for Service Discovery

Tom Leclerc*, Adrian Andronache†, Laurent Ciarletta*, Steffen Rothkugel‡

* MADYNES - INRIA Lorraine, Nancy,France. Email: {tom.leclerc, laurent.ciarletta}@loria.fr
† CRP Henri Tudor, Luxembourg. Email: adrian.andronache@tudor.lu
‡University of Luxembourg, Luxembourg. Email: steffen.rothkugel@uni.lu

***Abstract*—Service discovery is one of the most fundamental building blocks of self-organization. While mature approaches exist in the realm of fixed networks, they are not directly applicable in the context of MANETs. We investigate and compare two different protocols as basis for service discovery, namely OLSR and WCPD. OLSR is a proactive routing protocol while WCPD is a path discovery protocol integrating node and link stability criteria. Two conflicting objectives of service discovery are the coverage of service queries together with the required bandwidth. Simulations are performed based on a setting in a city center with human mobility. We show that OLSR outperforms WCPD in terms of coverage. Due to its proactive nature, however, bandwidth consumption is high. WCPD on the other hand is much more bandwidth efficient, but at the cost of lower coverage. Finally, we motivate employing OLSR on top of an overlay topology maintained by WCPD. This fosters stability while reducing overhead and keeping coverage high. As a first step towards a hybrid protocol, we aim at increasing the stability of the communication paths. To do so, an adaptive approach is used, which increases the robustness of the network topology structures.**

***Index Terms*—Mobile Networks, Clustering, Topology Stabilization, Service Discovery**

I. Introduction

In this paper we consider large Mobile Ad hoc NETworks (MANET) where the wirelessly connected devices communicate without any infrastructure with each other. In order to provide ad hoc networks with useful, user friendly and interesting features service discovery should be provided. Service discovery facilitates resource/data/multimedia sharing or for example ad hoc/situated games, furthermore it permits to take full advantage of the dynamic networks specificities.

The goal of service discovery is mainly to find services provided by other nodes in the network in an automated way and use them by knowing a basic set of information. Initially, service discovery protocols were designed for wired networks and most services were simple services, like for instance printing services. Not every node can or wants to achieve a given service. For example to print, a node doesn't need to be connected directly to the printer. Hence just by using the service provided by the node that is actually connected to the printer is enough to be able to print. In the last years a wide range of services became popular, like music sharing, game services or gateway services providing Internet access. Without infrastructure, as in ad hoc networks, the need to automatically, hence not manually which would be to complicated, discover services, that the network offers is even more crucial than in classical wired networks as no central information is available. Service discovery is even more indispensable for nodes with limited capabilities, which want to use a service without having the capability to host or run it by themselves. In ad hoc networks nodes, and the services they provide, can come and go so that topology changes all the time. These topological changes have to be reflected on the service discovery architecture.

In wired network a service failure is mostly due to a service inherent problem while in ad hoc networks topology causes most of the service failures.

In mobile ad hoc networks, just finding a service that suits best the user's and application's requirements is merely sufficient. In today's service-rich and growing networks, what matters is finding the best service that also optimizes part or all of the following requirements: the hop distance, stability, availability, effectiveness, etc. To enable these requirements an additional requirement which is a topological structure seems imperative.

We consider topology oriented protocols where some nodes have higher responsibilities like for instance relaying, grouping or disseminating messages from other nodes. Taking the topology building techniques from these protocols for service discovery protocols, allows us to have an efficient dissemination of service information and enables us to take advantage of the higher responsibility nodes. The higher responsibility nodes, also called directories in service discovery, store, forward or query service information for other nodes.

This paper is based on the work published at UBICOMM 2008 [1]. We investigate and compare the performances of the two topology conscious protocols OLSR [2] and WCPD [3], in regards to their topology architecture, for service discovery achievements. As the capabilities of the devices in ad hoc networks are always growing but still heterogeneous, from low capacities to very high, we consider a full range of services from simple classical printing services to advanced multimedia services. We present a hybrid approach using OLSR on top of WCPD and, as first step towards it, analyze a mechanism for stabilizing the cluster topology.

II. Related Work

As stated before, most of the service discovery protocols designed for wired network, like SLP [4], JINI [5], or UPnP [6] do not take into account any topology information.

Several discovery mechanisms can be implemented and mixed in service discovery protocols: active/passive discovery, directory or directory-less discovery. Active discovery means nodes broadcast a request for a service in the network and receive one or more answer from the service provider matching the request. Passive discovery means service providers periodically announce their services to all the nodes in the network. To reduce broadcasting in the network from many nodes, eventually resulting in massive flooding of the network, directory nodes are used. These nodes are elected by the surrounding nodes and are responsible for the electing set of nodes. Once elected, they store service announcements and corresponding service information, handle queries of their "slaves", hence reducing considerably the load of the network and the non-directory nodes.

Allia [7] is a peer-to-peer caching based and policy-driven service discovery framework. It removes the leader election problem by enabling every node to be self-sufficient. Every node creates alliances with other nodes and uses local policies for forward and caching decisions. A node knows which nodes are in his alliance, but it does not know in which alliances it is included from other nodes. As Allia does not take into account the network topology it does not fit our previously stated requirements.

Others propose to take partial aspects of the topology into account like in [8] and [9], where both use a multicast topology for the service discovery which is given by the network layer. Unfortunately the use of multicast induces a large number of control messages, which also does not suit our requirements.

The most interesting approaches for our work are the ones that take advantage of network topology to disseminate service information efficiently.

OLSR (Optimized Link State Routing) is well known as an ad hoc routing protocol but it is also a popular choice for service discovery architectures, mainly as an underlying connectivity provider. In [10] and [11] the OLSR protocol is used to encapsulate the service discovery messages. Furthermore in [12] the bordercasting, which is the "Multipoint Relay (MPR)" mechanism of OLSR, is used to efficiently flood the network.

Another interesting architecture is the Hierarchical OLSR [13] (HOLSR) which actually is not a service discovery protocol, but does address our problem of disseminating information through ad hoc networks efficiently.

The other type of topology we are taking into consideration is the cluster topology. Although in service discovery the cluster topology can be referred as service discovery with directory. The service discovery directories correspond to the clusterheads of the cluster architecture. Directories are elected on various criteria, like for instance node coverage.

A good example is Scalable Service Discovery for MANET [12] which is a distributed central directories discovery architecture. Directories are responsible for caching the service descriptions, advertising their presence to nodes within their vicinity and handling their service requests by checking the local cache or forwarding the query to other directories. The election of the directories is done on the fly and the main election criterion is the node coverage. To exchange the directory profiles they use bloom filters and "bordercast" (using MPRs) it in the two-hop neighborhood. However since the selection of the directory nodes relies on the node coverage can lead to problems. For example, superfluous elections occur when a nearby coming node traverses the network and obtains a high node coverage at one particular moment, but disconnects because of his mobility shortly after being elected, thus inducing a new election.

III. Topology protocols

This section briefly describes the protocols, OLSR and WCPD used in our experiments to find a good suited topology for service discovery. We choose to compare OLSR and WCPD because both build well known topology architectures. On one hand OLSR builds a tree topology and on the other hand WCPD builds a star topology.

A. OLSR

The Optimized Link State Routing Protocol (OLSR) is a well known routing protocol designed for ad hoc networks. It is a proactive protocol; hence it periodically exchanges topology information with other nodes of the network. One-hop neighborhood and two-hop neighborhood are discovered using Hello Messages (similar to the beacon message). The multipoint relay (MPR) nodes are calculated by selecting the smallest one-hop neighborhood set needed to reach every two-hop neighbor node. The topology control information is only forwarded by the nodes which are selected as MPR. Every node possesses then a routing table containing the shortest path to every node of the network. OLSR enables efficient flooding of the network by building a Tree like topology for every node from a source (Figure 1).

B. WCPD

The Weighted Cluster-based Path Discovery protocol (WCPD) is designed to take advantage of the cluster topology build by Node and Link Weighted Clustering Algorithm (NLWCA) [14] in order to provide reliable path discovery and broadcast mechanisms in mobile ad hoc networks (Figure 2).

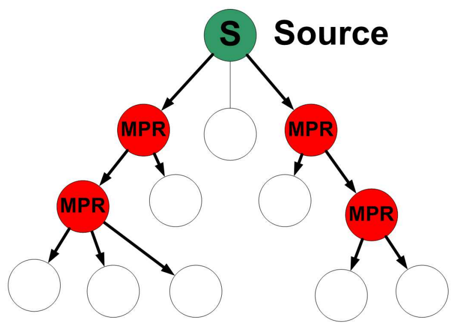

Fig. 1: OLSR topology for one source node in particular.

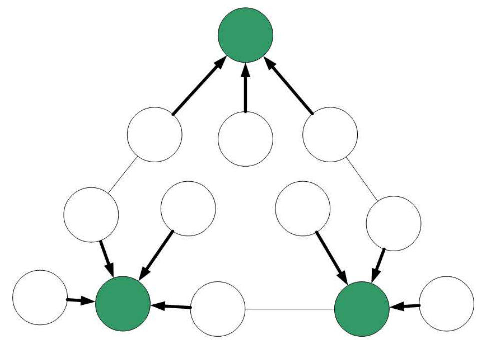

Fig. 2: WCPD cluster topology. The clusterheads are connected by multi-hop paths, which are used for inter-cluster information exchange.

NLWCA organizes ad hoc networks in one hop clusters by using only information available locally. Each device elects exactly one device as its clusterhead, i.e. the neighbor with the highest weight.

The main goal of the algorithm is to avoid superfluous reorganization of the clusters, particularly when clusters cross each other. To achieve this, NLWCA assigns weights to the links between the own node and the network neighbor nodes. This weight is used to keep track of the connection stability to the one-hop network neighbors. When a link weight reaches a given stability threshold the link is considered stable and the device is called stable neighbor device. The clusterhead is elected only from the set of stable neighbors which avoids the re-organization of the topology when two clusters are crossing for a short period of time.

WCPD discovers nearby stable-connected clusters in a pro-active fashion. For the nearby clusterheads discovery algorithm, WCPD uses the beacon, which is a periodically broadcasted message used in ad-hoc networks to detect devices in communication range.

WCPD runs on each network node and requires solely information available locally in the one hop neighborhood. The algorithm uses information provided by NLWCA: the set of stable connected network neighbor nodes and the ID of the own clusterhead. NLWCA also propagates by beacon the own weight and the ID of the current clusterhead. Besides the information provided by NLWCA, the WCPD protocol uses the beacon to disseminate the list of locally discovered nearby connected clusterheads.

By doing so, every node has the following information about each stable one hop neighbor: its clusterhead ID and the ID set of discovered clusterheads and the respective path length. After the data of all stable one hop neighbors is checked, the set of discovered nearby clusterheads and the path length is inserted into the beacon in order to propagate it to the one hop neighborhood.

The WCPD broadcasting algorithm is simple and easy to deploy: the broadcast source node sends the message to the clusterhead, which stores the ID of the message and broadcasts it to the one hop neighborhood. After that, it sends it to all nearby clusterheads by multi-hop unicast and to the own subheads by unicast. The inter-cluster destination nodes repeat the procedure except that the message source clusters are omitted from further forwarding. Additionally the information about the ID of the broadcast messages and their sources is stored for a given period of time to avoid superfluous re-sending of the message.

The protocol sends the broadcast message to nearby clusters connected by stable links in order to disseminate it to the network partition. Nevertheless the message also reaches crossing clusters since the broadcasts are received by all nodes in the one-hop neighborhood of local leaders. This increases the chance that the message reaches a high number of nodes in the mobile network partition.

C. Disseminating Messages

As our comparison relies on the information dissemination of both OLSR and WCPD, we furthermore compare both message dissemination mechanisms. When following the flow of a disseminating message, the topological structures, tree and star, of both protocols are highlighted.

The tree topology of OLSR is pointed out in Figure 3. A message sent from a source traverses the network by being forwarded only from the MPRs calculated by OLSR. As OLSR assures the full coverage of the network with the MPR selection, the messages reaches every node in the network.

The star topology is revealed in WCPD on Figure 4. Here a message from a source S (in this case a slave node) is first sent to its clusterhead B. Clusterhead B then one-hop broadcasts the messages to its slaves and multi-hop unicasts it to the nearby clusterheads A and C. Upon receiving the messages clusterheads A and C start the same procedure; broadcast to the slaves and unicast to nearby clusters (omitting source cluster). Thus every node (clusterheads and slaves) will receive the message. However nodes that are not considered as stable (e.g. fast moving nodes) might not receive the message

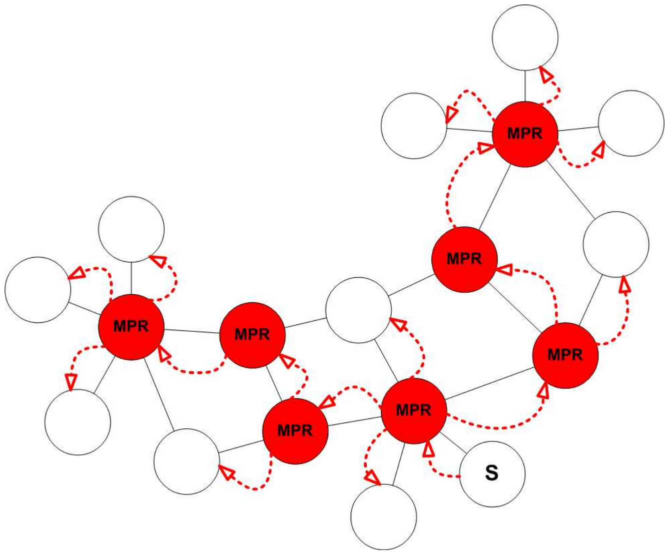

Fig. 3: OLSR message dissemination through a network.

unless they are in the direct neighborhood of a clusterhead that is broadcasting the message (i.e. intended to its slaves).

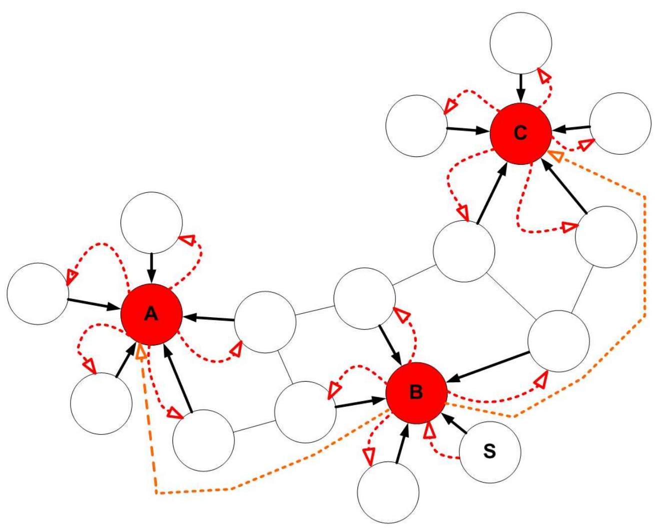

Fig. 4: WCPD message dissemination through a network.

IV. Experiments

In order to determine the best suited topology for our service discovery protocol, we implemented both protocols on the top of the JANE simulator [15] and performed several experiments.

A. Simulation settings

For the conducted experiments we used the Restricted Random Way Point mobility model [16], whereby the devices move along defined streets on the map of Luxembourg City for 5 minutes (Figure 5). For each device the speed was randomly varied between [0.5;1.5] units/s. At simulation startup, the devices are positioned at random selected crossroads and the movement to other crossroads is determined by the given random distribution seed. For the experiments a number of 10 different random distribution seeds were used in order to feature results from different topologies and movement setups.

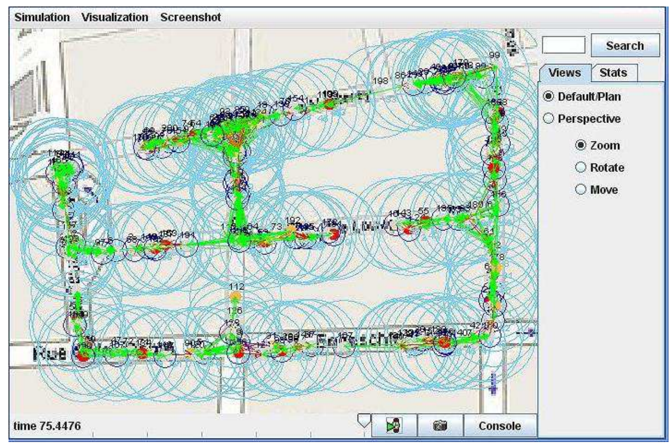

Fig. 5: JANE simulating the protocols on 100 devices. The mobile devices move on the streets of the Luxembourg City map. The devices move with a speed of 0.5 - 1.5 m/s.

For the used mobile environment where nodes move with low speeds between 1.8 and 5.4 km/h the NLWCA link-stability threshold is set on 2. Simulations were done to determine both the used bandwidth in order to build the topologies and the information dissemination performance of broadcasting on top of the two different topologies.

To build the MPR topology, OLSR exchanges the sets of one-hop neighbor nodes with every node in the communication range. Similar to OLSR, WCPD use the beacon to exchange the list of the discovered nearby-clusterheads with the one-hop neighbor nodes. To find out the network load produced during this phase, the size of both the one-hop neighbor sets and the size of discovered clusterheads were tracked every second of the simulation. In order to monitor the information dissemination performance and network load of the broadcasting mechanisms, a node was chosen to broadcast a message every 10 seconds during different simulation runs using different distribution seeds. The number of sent messages (i.e. broadcasts and unicasts) during the dissemination and the number of reached network nodes were tracked.

B. Results

The results in figures 6, 7 and 8 are illustrating the size of the exchanged node-ID lists at the respective point in the timeline.

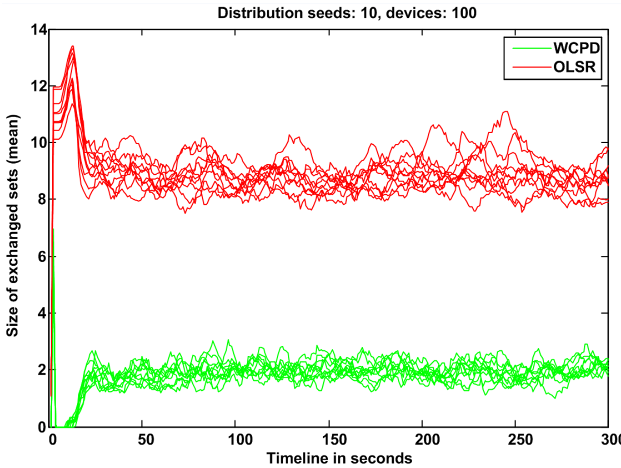

Fig. 6: Size of the sets exchanged per second in order to build the topology.

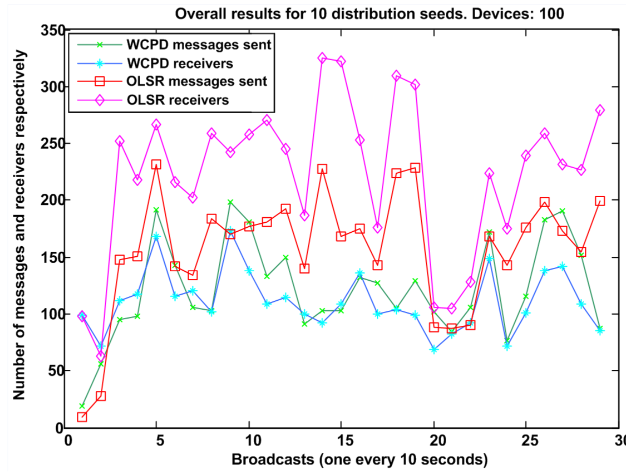

Fig. 9: Overall number of sent messages and node receivers respectively for 100 nodes.

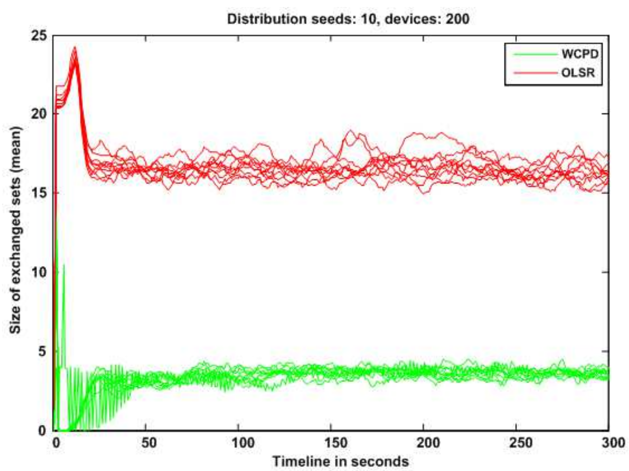

Fig. 7: Size of the sets exchanged per second in order to build the topology.

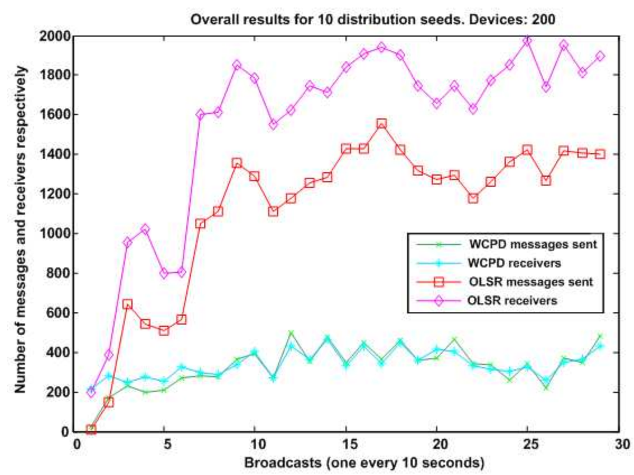

Fig. 10: Overall number of sent messages and node receivers respectively for 200 nodes.

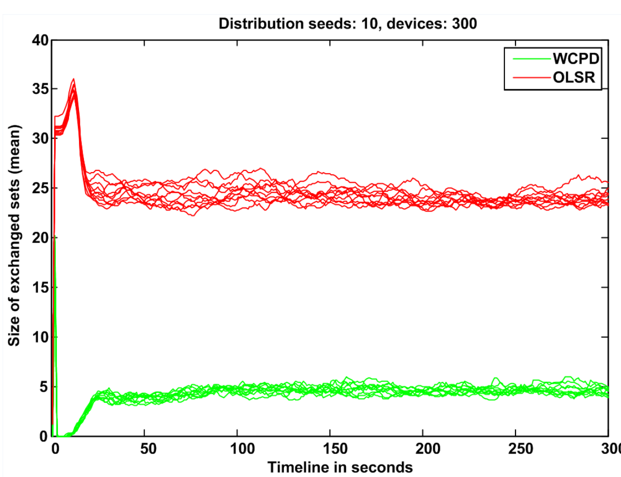

Fig. 8: Size of the sets exchanged per second in order to build the topology.

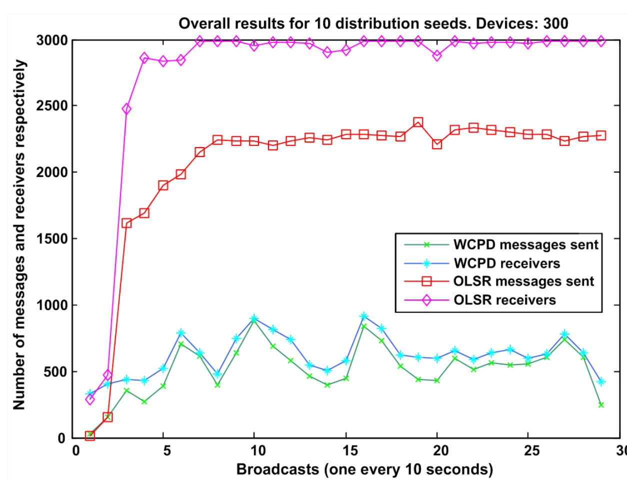

Fig. 11: Overall number of sent messages and node receivers respectively for 300 nodes.

To calculate the bandwidth used by the protocol, one needs to take into consideration the time interval used to periodically send the exchange messages (i.e. hello messages or beacons) and the size of the used node IDs (e.g. 32 bits for IPv4 addresses). This leads to formula 1 for a mean bandwidth B used in an IPv4 network where $|S|$ is the mean number of exchanged addresses and t is the time between the periodically exchanges:

$$B = \frac{|S| \times 32}{t} \text{bits/sec} \tag{1}$$

The results illustrated in figures 6, 7 and 8 show that OLSR uses a higher bandwidth in both sparser and denser networks. This situation was expected since OLSR is exchanging the set of one-hop neighbors needed for the MPR nodes election.

In contrast to OLSR, WCPD only exchange the set of local discovered nearby clusterhead and sub-heads in order to discover stable paths between clusters in the network vicinity. The NLWCA protocol elects one clusterhead/sub-head in each one-hop neighborhood, which means that the number of clusterheads is a fractional amount of the number of nodes in the network.

The tracking results regarding the message dissemination performance and network load of the broadcasting protocols are presented in figures 9, 10 and 11. The overall results show that the broadcasting on top of the OLSR topology performs much better in terms of message dissemination than on top of the WCPD topology. The denser the network, the higher is the difference between both the number of sent messages and the number of receiver nodes.

V. A HYBRID APPROACH SOLUTION

OLSR broadcasting is based on flooding the network in an efficient way via the MPRs in such a way that messages reach all nodes already captured. Even in the presence of mobility, the broadcast will arrive at a high number of nodes. In contrast to that, the WCPD approach aims at spreading the messages between topology structures that are considered to be connected in a stable way. Especially in the presence of mobility, the stability threshold might not be reached by all nodes, which might result in a smaller number of broadcast receivers.

We propose to overlay both topologies—in this context for service discovery—by employing the OLSR MPR algorithm on top of the WCPD cluster topology.

In this hybrid approach clusterheads are used as service discovery directories. The discovery of nearby directories in turn is facilitated and maintained by the WCPD protocol. The communication paths between the directories used to exchange service discovery information are maintained by OLSR. Thus, the OLSR protocol has to establish the MPR topology only between clusterheads, which dramatically reduces the required communication load. Additionally, the OLSR topology on top of the cluster topology will result in optimized inter-cluster communication paths.

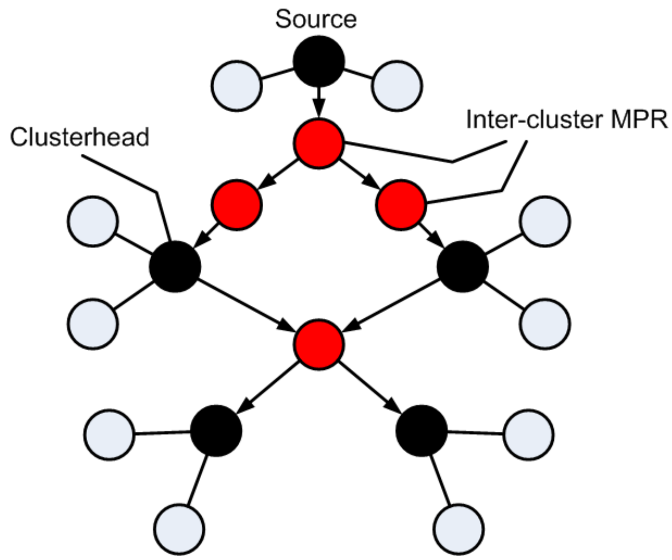

Fig. 12: A hybrid architecture where the OLSR MPR protocol is used to connect nearby clusters discovered by the WCPD protocol.

A. Stabilizing the cluster topology

The performance of WCPD depends on the stability of the underlying NLWCA topology. In order to increase the reliability of WCPD, the robustness of the cluster topology has to be increased. This section presents the adaptive NLWCA approach, which is a first step towards the hybrid OLSR-WCPD communication protocol.

The NLWCA protocol uses a link stability threshold (LST) in order to decide if a communication link to a neighbor mobile node is stable or not. Simulation results showed that low thresholds are best suited for networks with low mobility. For instance, such networks can be formed by device of users that are in school rooms, cinemas, restaurants, pubs and so on. In such settings a low LST enables the NLWCA to organize fast the local devices, thus reducing the number of elected clusterheads.

In settings where the network nodes are moving around more often and faster, a higher LST is better suited. This allows stable connected clusters to cross each other without to be re-organized by NLWCA. Such networks are created by devices of users for instance at train and subway stations, on the streets of big cities, in shopping malls and so on. The higher LST avoids the organization of crossing nodes but as consequence it increases the number of clusterheads in the mobile network.

The value of the LST has a critical impact on the NLWCA topology. If the LST is to low then the topology is unstable, which means that nodes re-affiliate to new clusterheads very often. This triggers additional network communication and also decreases the robustness of the stable inter-cluster paths. On the other side, a LST that is to high for the given mobility setting leads to election of superfluous clusterheads in the mobile network.

In real mobile environments the network nodes often change their position and the mobility setting. Besides this, scenarios with mixed mobility are common in reality. For instance the nodes in a restaurant have a low mobility and a low LST is best suited. But some nodes in the restaurants that are near to the street might be in communication range of nodes passing by on the street. Thus, these nodes are in a mixed mobility area. Such nodes require a higher LST than the nodes positioned more back in the restaurant in order to avoid superfluous re-affiliations with nodes on the street. This example shows that a constant LST is not the best suited approach for network models with different mobility settings.

To avoid the drawbacks brought by a constant LST, the NLWCA protocol is augmented by a mechanism that allows the change of the LST during runtime. Thus, the protocol is able to adapt the threshold to the given network mobility in order to increase the topology stability.

B. *The adaptive NLWCA protocol*

The augmented NLWCA protocol enables each network node to maintain an own link stability threshold. The LST is inserted into the network beacon in order to make it known to the neighbor nodes. When two nodes enter the communication range, the higher of the two LSTs is chosen to be used by both nodes for the link between them. For instance, a node from a high mobility area with a high LST might pass a low mobility area where the nodes have low LSTs. In this case NLWCA uses the high LST of the passing node for the links between it and the nodes in the low mobility area (Figure 13). Thus, a cluster affiliation of the passing node is avoided.

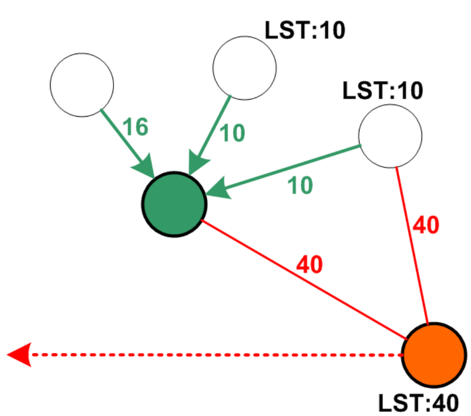

Fig. 13: The higher LST is used by the nodes for the communication links between them. This avoids the cluster affiliation of the node passing by.

The implemented adaptive NLWCA protocol adapts the LST of the nodes by following the listed rules:

1) The LST takes values between 3 and 600 seconds.
2) Monitor the one-hop network neighborhood for a time span of 10 seconds.
3) If stable links are disconnected during the time span then increase the LST by 3.
4) If no link (stable or unstable) is disconnected during the time span then decrease the LST by 1.
5) Updates of a node LST triggers an update of the LSTs of all of its links.
6) Already stable links remain stable even if the LST is increasing. This protects already stable structures from re-organization.
7) Go to rule 2.

Note: All values used might be changed in future work in order to increase the performance of the adaptive mechanism.

VI. Simulation experiments and results

The goal of the first simulation experiments was to keep track of how NLWCA performs by adapting the LST under mixed mobility settings. In order to do so, a mobility scenario with three mobility settings was created (Figure 14). The first area is a 400 meters long stripe that represents a street. The half of the network nodes used in the simulation randomly move along the street area from one end to the other end with a random speed between 0.5 and 2.5 meters/second. These nodes create the high mobility area of the simulation scenario. Along the street, five areas with low mobility are created. These areas represent restaurants, pubs or other places where people might spend some time. Half of the network nodes are randomly distributed on these areas and they are not leaving the areas for 15 minutes of simulation, thus creating low mobility network settings. The nodes near to the street area are in communication range of the nodes passing by. Thus, these nodes are in a mixed mobility area.

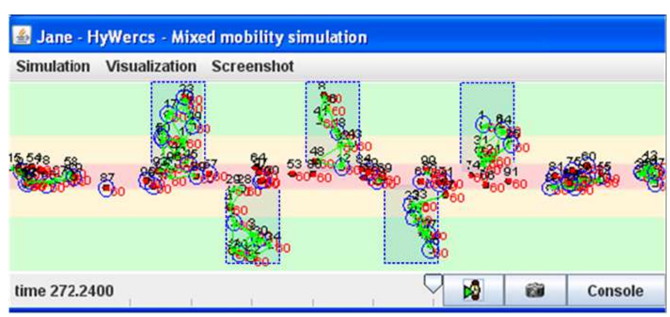

Fig. 14: Simulation scenario with one high mobility area and five low mobility areas.

The sending radius of the nodes was set to 20 meters. In order to compare the adaptive approach with the previous static protocol, the simulation runs were repeated with LST values of 3, 30, 60, 120 and 180 seconds. Each simulation setting was conducted with 10 different distribution seeds. The first sets of experiments were done with a number of 100 mobile nodes.

Figure 15 shows the mean number of elected local leaders during 15 minutes simulation time. A local leader is a clusterhead or a sub-head since both are used by WCPD for inter-cluster path discovery. A low number of local leaders is advantageous since it reduces for instance the backbone communication and the inter-cluster information exchange overhead.

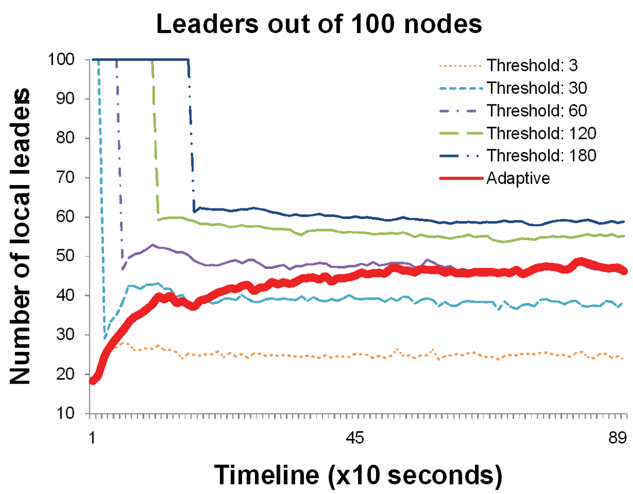

Fig. 15: Number of elected local leaders out of 100 nodes during 15 minutes of simulation.

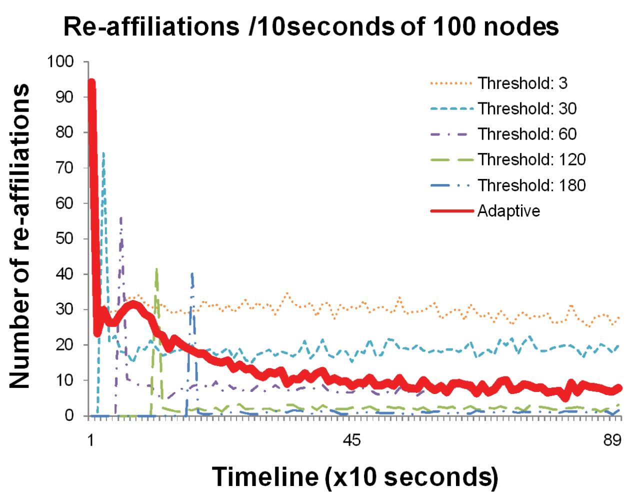

Fig. 16: Number of cluster re-affiliations per 10 seconds during 15 minutes of simulation.

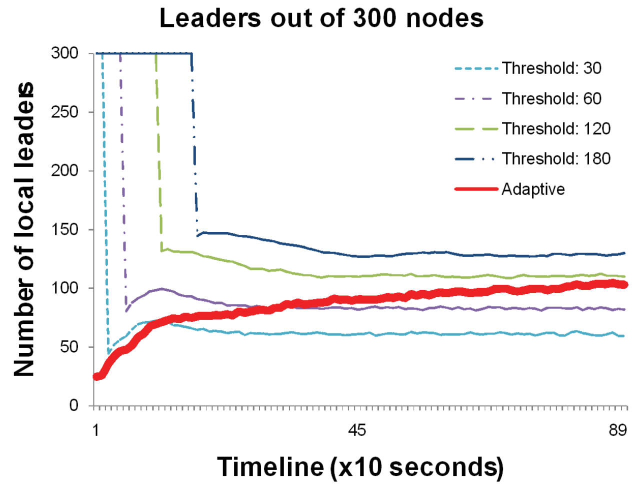

Fig. 17: Number of elected local leaders out of 300 nodes during 15 minutes of simulation.

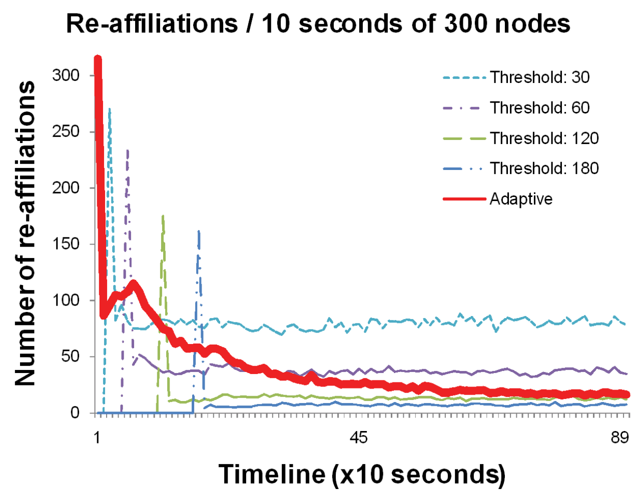

Fig. 18: Number of cluster re-affiliations per 10 seconds during 15 minutes of simulation.

The results in figure 15 show that the lower thresholds lead to a low number of elected local leaders. This is due to the fast organization of the mobile nodes when using a low LST. The drawback of this setting is that crossing clusters are not protected from re-organization. This can be observed in Figure 16, which shows the number of cluster re-affiliations tracked every 10 seconds. A re-affiliation means that a node changed its clusterhead, thus affiliating to another cluster. This induces network communication overhead as well as inter-cluster paths losses or re-configurations.

The lowest LST of 3 seconds triggers a mean value of 30 re-affiliations per 10 seconds compared with 1 re-affiliation per 10 seconds triggered by the 180 seconds LST. This means that the high LST leads to more robust cluster structures. The drawback of the high LST is that it produces a high number of local leaders like Figure 15 shows. Besides this, in low mobility areas such high LSTs are not necessary and lead to a slow cluster organization.

The adaptive NLWCA protocol acts as expected during the simulations. In the beginning, it starts with a low LST, which triggers a high number of re-affiliations by organizing the network in a small number of clusters. Since NLWCA aims to increase the stability of the cluster structures it increases the LST on the nodes in high mobility areas. This leads to a higher number of local leaders (Figure 15) but it highly reduces the number of re-affiliations (Figure 16), thus increasing the robustness of the topology structures.

The same experiments settings were used in simulations with denser networks of 300 mobile nodes. The results are illustrated in Figures 17 and 18.

The behavior of the adaptive protocol in networks with 300 nodes is similar to the one observed in networks with 100 nodes. By increasing the LST of the nodes in high mobility areas, the adaptive NLWCA protocol increases the number of local leaders, thus decreasing the number of re-affiliations.

The results of the conducted adaptive NLWCA simulation experiments are very promising. Nevertheless, experiments with a higher number of network environment scenarios are planned as future work in order to optimize the parameters of the adaptive protocol.

VII. Conclusion & Future Work

The simulation results show that between the two analyzed approaches, the one based on OLSR is the better choice in order to reach as many nodes as possible by broadcasting for instance service discovery queries. This protocol highly outperforms in terms of broadcast receivers the WCPD approach that fosters the communication between nearby clusters considered to be stable-connected. On the other side, the network load produced by OLSR to build the topology is much higher compared to the one of the WCPD protocol. Besides that, services discovered on nodes in the network vicinity are more valuable than the ones on nodes topologically far away. The multi-hop path to a service host can be easily lost in mobile environments due to the movement of the nodes or network partitioning. In conclusion the OLSR broadcasting approach has the advantage of reaching a much higher number of nodes than WCPD but at the cost of high network overload for the topology maintenance.

In future work we aim to combine the two protocols in a synergetic way by building clusters of stable connected nodes and using the OLSR topology on top of the cluster topology. Thus, a better inter-cluster path discovery and loop-free broadcasting mechanism may be provided at a low network load used for topology maintenance. This will enable the service discovery protocol to take advantage of stable paths to service hosts in the vicinity and at the same time to reach a high number of network nodes by broadcast.

In mobile network environments devices might experience various mobility settings. To increase the stability of the cluster structures NLWCA was augmented to adapt the link stability threshold to the given network mobility. Experiment results show that the middleware successfully reduces the cluster re-affiliations of the mobile devices, thus increasing the robustness of the network structures.

As next step, the hybrid protocol will be deployed and analyzed on top of the robust cluster topology.

References

[1] T. Leclerc, L. Ciarletta, A. Andronache, and S. Rothkugel, “Olsr and wcpd as basis for service discovery in manets,” in *UBICOMM ’08: Proceedings of the 2008 The Second International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 184–190.

[2] “Optimized link state routing protocol (olsr), rfc3626,” United States, 2003.

[3] A. Andronache and S. Rothkugel, “Hytrace backbone-assisted path discovery in hybrid networks,” in *CTRQ ’08: Proceedings of the 2008 International Conference on Communication Theory, Reliability, and Quality of Service*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 34–40.

[4] J. Veizades, E. Guttman, C. Perkins, and S. Kaplan, “Service location protocol,” 1997.

[5] Jini technology home page. http://www.sun.com/software/jini/.

[6] Universal Plug And Play Forum. http://www.upnp.org/.

[7] e. a. Ratsimor O., “Allia: Alliance-based Service Discovery for Ad-Hoc Environments,” in *ACM Mobile Commerce Workshop*, 2002.

[8] S. e. a. Helal, “Konark a service discovery and delivery protocol for ad-hoc networks,” 2003.

[9] U. C. Kozat and L. Tassiulas, “Service discovery in mobile ad hoc networks: a field theoretic approach,” 2005.

[10] L. Li and L. Lamont, “A lightweight service discovery mechanism for mobile ad hoc pervasive environment using cross-layer design,” *Pervasive Computing and Communications Workshops, IEEE International Conference on*, 2005.

[11] e. a. Jose Luis Jodra, “Service discovery mechanism over olsr for mobile ad-hoc networks,” *Advanced Information Networking and Applications, International Conference on*, 2006.

[12] F. Sailhan and V. Issarny, “Scalable service discovery for manet,” in *PERCOM ’05*, Washington, DC, USA, 2005.

[13] L. Villasenor-Gonzalez, Y. Ge, and L. Lament, “Holsr: a hierarchical proactive routing mechanism for mobile ad hoc networks,” *Communications Magazine, IEEE*, vol. 43, no. 7, pp. 118–125, July 2005.

[14] A. Andronache and S. Rothkugel, “Nlwca node and link weighted clustering algorithm for backbone-assisted mobile ad hoc networks,” in *ICN ’08: Proceedings of the Seventh International Conference on Networking*. Washington, DC, USA: IEEE Computer Society, 2008, pp. 460–467.

[15] D. Gorgen, H. Frey, and C. Hiedels, “Jane-the java ad hoc network development environment,” *Simulation Symposium, Annual*, 2007.

[16] L. Blažević, S. Giordano, and J.-Y. Le Boudec, “Self organized terminode routing,” *Cluster Computing*, vol. 5, no. 2, pp. 205–218, 2002.